STATISTICS FOR COMPUTER SCIENTISTS

Dr. Walter S. Szalajka
Box 1111
Lewis University
Romeoville, Illinois 60441

Introduction

In Departments of Mathematics it is common to offer a a two semester course sequence of probability and statistics that is calculus based and is primarily taken as an elective or requirement by mathematics majors. This sequence is well-established and many texts exist for it. The common difficulties with this sequence are: (1) enrollments are low, (2) the courses do not seem to attract other majors with a calculus background (science students and now large numbers of computer science students), (3) there is usually no computer usage in the courses (for example, usage of computer statistical packages, (4) the courses have no computer programming component, (5) the material in the courses sometimes seems geared for students intending to enter graduate schools of statistics, whereas most of the students in the course will probably immediately enter private industry. There is usually too much probability theory taught in the first course and too few students remain to elect Probability and Statistics II where very important applied statistical material is taught. Something needs to be done to this calculus based statistical sequence to make it more attractive and practical to a wider population of students while maintaining a mathematical statistics component. This paper describes a statistics course that is required of computer science majors which replaced the traditional Probability and Statistics I and II course sequence at Lewis University. The writing of this paper was motivated by the general discussion about statistics in the mathematical sciences program at the American Mathematical Society Short Course on Modern Statistics: Methods and Application, San Antonio, January 7-8, 1980. Ideas for the course also originated at UCLA Short Courses [3,10].

Statistics for Computer Scientists

A knowledge of statistics at various levels is important for many undergraduate majors; elementary statistics is a common experience in everyday life. Non-calculus based statistics courses serving business, economics, psychology, sociology, and education are not the subject of this paper, but ideas can be taken from here to revitalize those courses with a computer component. A knowledge of statistics is also important for computer science majors: common statistical tests have all been computerized and are found in various packages in computer installations; statistics is so pervasive in our society that a knowledge of statistics can be an important additional skill when a computer science major is seeking an entry level position in the data processing industry; and "a basic knowledge of statistics is essential to almost all areas of professional work in computer science" [5]. Statistical tests provide a source of substantial programming assignments in which the computer science major can experience working with large, real datasets. Computer science majors usually have a calculus background and need a more substantial course in statistics than the non-calculus based statistics service courses. They need some exposure to probability theory and a survey of statistical tests in common use, but do not have to become statisticians. They have time for only a one semester course in statistics. Thus, the usual Probability and Statistics I and II course sequence does not meet the needs of these computer science majors. The following represents the necessary adjustment in Probability and Statistics I and II to meet the needs of an expanding computer science major population while maintaining a mathematical statistics course appropriate for mathematics majors.

Mathematics/Computer Science 315 (4 semester hours credit) was taught by the author in the fall of 1977 (30 students) and the fall of 1979 (28 students). The prerequisite is Calculus II and a knowledge of computer programming (FORTRAN or BASIC). This course is required of computer science majors and can be elected by mathematics majors. The course outline appears in Appendix I. After a week on elementary descriptive statistics five weeks are spent on elementary probability theory. Statistical tests are then covered one week per test for the remainder of the course (cookbook explanations first followed by various levels of theoretical justification).

The text for the course in 1977 and 1979 was not ideal because it is not computer based [4]. However, it is calculus based, gives a short survey rather than an overwhelming amount of probability theory, and then surveys common statistical tests in both a practical and theoretical way. Attempts were made in the past to find a more suitable text to meet the course specifications but they did not succeed. A non-calculus based text having a computer component was seriously considered in 1979 but not adopted [6]. Recently, it was decided to use a new text for the fall of 1981 that is calculus and computer based [2].

Computer Component

The computer available to students in the course was a PRIME 300 Timesharing Minicomputer housed in a Computer Laboratory administered by the Department of Mathematics and Computer Science and dedicated to academic computing. All program development work in the course is done interactively on six Hazeltine 1500 Visual Display Terminals and six DECwriter II Printer Terminals. The classroom and faculty offices for the Statistics course are adjacent to the Computer Laboratory, allowing terminals to be brought into the classroom for demonstration purposes. Canned statistical packages are available on the computer and two statistical calculators are also available on the computer and two statistical calculators are also available in the Computer Laboratory (a Canon F-20P Statistical Calculator and Hewlett Packard Model 81 Business Statistical Calculator). Surprisingly, students do make use of statistical calculators in doing homework, on tests, and in checking computer programming assignment test data. A very convenient computer environment is important for this course. Since it is a statistics course and not a computer programming course, this environment minimizes computer difficulties associated with the course.

The computer programming assignments for the course are given in Appendix II. In the computer science major program, many small programming assignments are given in lower division courses [9]. In upper division courses an attempt is being made to simulate large software development projects as much as possible. The preferred language for the programming assignments is FORTRAN because of its widespread use in writing commercial statistical packages. BASIC is acceptable because of its widespread use on microcomputers and as an ongoing experiment to see how much statistical programming can conveniently be done in BASIC. Language limitations are quickly encountered when coding statistical programs in either FORTRAN or BASIC. This naturally leads to a discussion of which language is most appropriate to code statistical programs, a topic from a programming languages course. All the programs are to be written with an unsophisticated user of the program in mind. A user should be able to use a student program with little or no explanation and the program should be documented well for an interactive user. A well-documented modular programming style is also stressed and found to have great value when using parts of previous programs to do current programs. Bad data checks are emphasized, and checking that output is sensible is also stressed. Students are encouraged to save their programs and add them to their personal portfolios when job hunting.

Programming Assignment 1 is an elementary data analysis assignment. Two very large datasets of final exam test scores are used. One dataset contains bad data (for example, 110% and -20%). Students must decide which appropriate formulas to use (if, in fact, formulas exist). At an appropriate time, substantial references concerning this problem can be given [1,11]. Students can experience the following issues during the programming assignments: the problem of static dimensioning of variables in FORTRAN and knowing the number of items in the dataset; the problem of keeping file names of the dataset independent of the program; minimizing the number of times the data is read (an efficiency consideration); what sorting algorithm to use for a possible median calculation (a topic from a data structures course); internal documentation; external documentation for an interactive user of the program; checking for bad data; making sure the output is reasonable; and the value of a modular style. These issues can be addressed when the programming assignments are returned. Various individual solutions to these problems can be very instructive for the class as a whole.

Programming Assignment 2 is a probability theory assignment. Students must have a knowledge of the density function of a continuous random variable and how it is used to get probabilities in the non-trivial case of the normal distribution. Students do really enjoy the challenge of trying to duplicate the values for the probabilities of a normally distributed random variable that they can get from a table in their textbook or from a statistical calculator, since there is an aura of real practicality in this. This can also be considered as a numerical analysis/numerical methods course assignment. Accuracy of the results of a student program and the domain of numerical analysis can optionally be discussed. Appropriate techniques to do the assignment (Simpson's Rule, the Trapezoid Rule, or merely using rectangle approximations) are readily available to students in their calculus books.

Programming Assignment 3 is a straightforward programming assignment of complicated formulas. Students can use modules from Programming Assignment 1 to do this assignment. Intelligible output is emphasized. A confidence interval or a rejection with appropriate justification should be outputed meaningfully to a user. However, the following are the main issues in doing this program: (1) should a t-distribution table be stored and searched as needed (searching is topic from a data structures course); or (2) should normal distribution values approximating the t-value be stored and searched as needed; or (3) should t-values or normal values be generated as needed (see the optional part of Programming Assignment 2). Students make their own choices in solving this issue and the teacher comments on all of the possible choices when the assignments are returned.

Programming Assignment 4 is also a very straightforward programming assignment of complicated formulas. Modules from previous programs can profitably be used here to minimize programming duplication. A dataset of very non-linear data (clearly non-linear when graphed) is also included. However, students do not usually discover the non-linearity of the data until after the assignment. This assignment actually serves as a starting point for discussions of interactive multilinear regression computer packages and the value of graphics (looking at a picture of the residuals rather than just their values).

Some commercial statistical packages are used in the course, but because of the substantial probability and statistics content and the very interesting computer programming assignments, these packages have not been used greatly in the course. Programming Assignments 1, 3, and 4 are done by the teacher using SPSS and BMD, with copies of the actual codes necessary to access these packages and their evaluation of the data given to the students when their programs are passed back. Comparing what the students have done and what these common statistical packages do (and at what cost) provides an opportunity for a fine classroom discussion.

Future Development of the Course

The most critical need for the course is a more appropriate textbook. The computer programming assignments seem appropriate to the course; more of the optional parts of the assignments could possibly be required in the future. More work could be done with important commercial statistical packages like SPSS and BMD. A term paper project doing a comparative study of such commercial statistical packages has been considered, using scarce published material as a model [7,8]. The current computer facilities for convenient interactive program development work in the course are adequate. However, computer-generated graphics capabilities (for histograms, graphs of residuals, and so forth) would be very advantageous in the future. A permanent computerized classroom environment in which the teacher, and possibly the students, will have terminals directly connected to the computer for classroom demonstration purposes would be helpful. The course content now is primarily parametric statistics, but non-parametric statistical work is being considered for inclusion.

The course has been successful thus far in becoming computerized and will clearly continue to develop!

Summary

During the past three years we have made a successful change from the traditional calculus based Probability and Statistics I course sequence to a one semester calculus based statistics course that is required of computer science majors and can be elected by mathematics majors. The course contains an introduction to probability theory and then surveys important statistical tests. The computer component of the course consists of four very substantial programming assignments. The programming assignments are done interactively on a conveniently located departmental timesharing minicomputer. In the future more work will be done in the course using statistical packages and graphics.

References

1. Chan, Tony F. and Lewis, John Gregg. Computing Standard Deviations: Accuracy. Communications of the ACM, September, 1979.

2. Groenveld, Richard A. An Introduction to Probability and Statistics Using BASIC. Dekker, New York, 1979.

3. Haming, Richard W. Statistical Concepts for the Computer Scientist. UCLA Short Course, July 30 - August 3, 1973.

4. Kreyszig, Erwin. Introductory Mathematical Statistics. Wiley, New York, 1970.

5. Ralston, Anthony and Shaw, Mary. Curriculum '78 - Is Computer Science Really that Unmathematical. Communications of the ACM, February, 1980.

6. Scalzo, Frank and Huges, Rowland. Elementary Computer-Assisted Statistics Petrocelli/Charter, New York, 1975.

7. Slysz, William D. An Evaluation of Statistical Software in the Social Sciences. Communications of the ACM, June 1974.

8. Schucany, W.R. and Minton, Paul D. A Survey of Statistical Packages. Computing Surveys, June, 1972.

9. Szalajka, Walter S. and Walch, Philip. Integrated Theory and Practice--an Approach to the First Computer Science Course. SIGCSE Bulletin, February, 1979.

10. Van Tassell, Dennis. Computer Statistical Packages. UCLA Short Course, May 17-21, 1976.

11. West, D.H.D. Updating Mean and Variance Estimates: an Improved Method. Communications of the ACM, September, 1979.

Appendix I

Mathematics/Computer Science 315 Statistics Course Outline
(four meetings per week)

Descriptive Statistics (one week) and Programming Assignment #1

Fundamental Concepts of Probability Theory (one week)

Probability Distributions (one week)

Parameters of Distributions and Discrete Distributions (one week)

Normal Distributions (one week) and Programming Assignment #2

Several Random Variables (one week)

Estimation of Parameters (one week)

Confidence Intervals (one week)

Hypothesis Testing (one week) and Programming Assignment #3

Chi-Square (one week)

Regression (one week) and Programming Assignment #4

Analysis of Variance (one week)

Correlation Analysis (one week)

Nonparametric tests (one week)

(In a sixteen week semester approximately two weeks remain for tests and reviews.)

Appendix II

Computer Programming Assignments

Programming Assignment #1: Write a computer program in FORTRAN or BASIC to access the following two test datasets and do the following: calculate and intelligibly print out the (sample) mean, (sample) variance, standard deviation, minimum, maximúm, and range of each dataset. You could also go on and calculate the median, mode(s), absolute, relative, and cumulative frequencies, output a histogram, standard error of the mean, skewness, kurtosis, z-score of the minimum, and z-score of the maximum. Use any appropriate formulas of your choice in this program. A "novice user" should be able to use and understand your program; document your program well!

Hand in a listing of your program, a listing of your data files, and the intelligible output of running your program on the given test datasets. (Two large test datasets follow; they could also be stored in the computer and the file names and format used could be given to the students.)

Programming Assignment #2: Let X be a normally distributed random variable with mean 0 and variance 1; calculate the following probabilities (namely, your program should request appropriate information to be INPUTTED and give back intelligible results):

$$P(-0.5 \leq X \leq 1)$$

$$P(-1.5 \leq X \leq 1.5)$$

$$P(0 \leq X \leq 2) \quad .$$

Use any appropriate technique of your choice in this program. We are essentially trying to reproduce Tables 3a and 3b of your text; namely, I can refer to your program rather than these tables when I want Normal Probability Results! Write the program in FORTRAN or BASIC.

Optionally, you can also consider the following extensions of the above assignment: (i) find z so that $P(X \leq z) = 95\%$; (ii) find $P(X \leq 1)$ and $P(X \geq 1)$; (iii) if Y is N(2,1) then find $P(-2 \leq Y \leq 2)$ and $P(Y \leq 0.5)$.

Programming Assignment #3: (a) Write a computer program to find a Confidence Interval for the Mean of a Normal Distribution with unknown variance; try your program out on Example 1 on page 179 of your text. Note that you can store a t-distribution table in your program, reproduce it as needed, or approximate the t-distribution with a normally distributed random variable. (b) Write a computer program to test the hypothesis $\mu_X = \mu_Y$ against the alternative $\mu_X \neq \mu_Y$ in the case of Normal Distributions with the same unknown variance; try your program out on Example 1 on page 211 of your text. Note that it is again all right to use the normal to approximate the t.

Programming Assignment #4: Find the regression line for each of the following two sets of data; also calculate the residuals in each case and interpolate and extrapolate as indicated with your regression line. (Two large datasets follow; interpolation values and extrapolation values are also given. The datasets could also be stored in the computer and the file names given to the students.)